Brandon Death 101017130
Matt Walker 100980836

Nov. 25, 2018.

Assignment #3 Answers

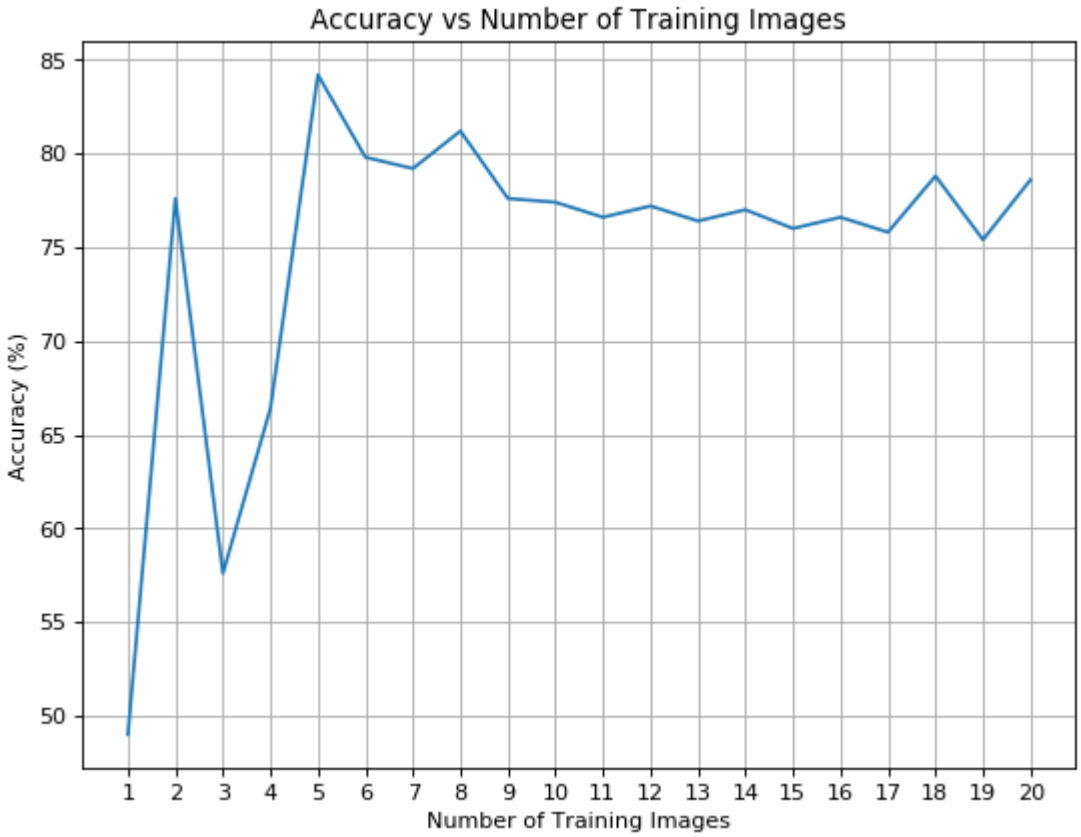
Question 1

a) For this question, we separated the images of ones and fives and converted them to binary images with 1 and -1 values. We then used a range of hand-selected initial training images from 1-10, and for each set we used the one-shot method to determine the initial weights of the basic Hopfield Network. These training images were then used as the "stored patterns". We then passed 500 test images into the network, and for each image we calculated the Euclidean distance to each stored pattern to determine which pattern it was closest to and then compared with the digit matched the test label. We then plotted the accuracy vs the number of training images, as shown below.

As seen from the graph above, after two images (one 1 and one 5), the accuracy drops off steeply to ~50% (random chance) as the stored patterns merge into a blur.

Brandon Death 101017130
Matt Walker 100980836
Nov. 25, 2018.

b) For this bonus question, we change the method of determining the weights of the network to the Storkey learning rule method. To do this, we implemented an algorithm written by Russell Richie (https://stats.stackexchange.com/questions/276889/whats-wrong-with-my-algorithm-for-implementing-the-storkey-learning-rule-for-ho). This improved the accuracy for a higher number of training images and peaked for 5 training images, as seen in the graph below.



Brandon Death 101017130
Matt Walker 100980836

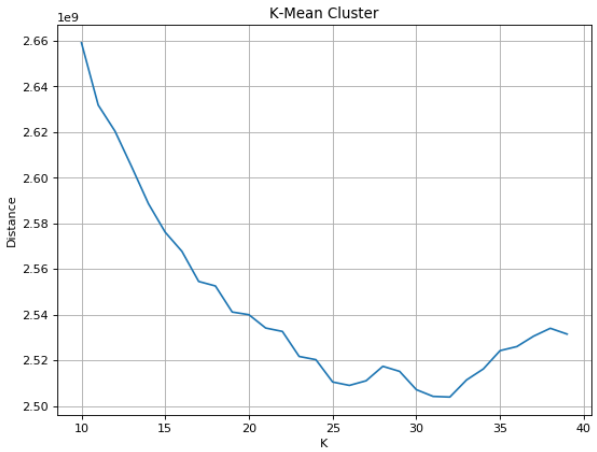
Nov. 25, 2018.

Question 2

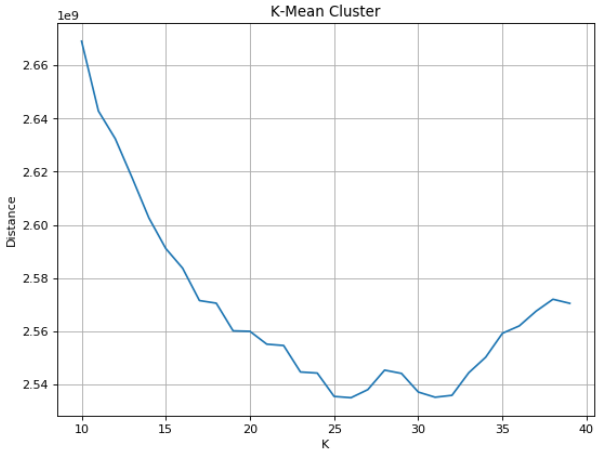
1. For this question, we built an RBF network with Gaussian functions as the RBFs. We used kmeans to cluster the images, and from this clustering we used the cluster centers as the mean of each corresponding Gaussian neuron. We also calculated the summed Euclidean distance of each point in the cluster to its corresponding center in order to determine the variance (and thus β) for the Gaussian functions. We used a range of k-values from 10-40 and plotted the distance of each point (training image) to its nearest cluster center + ck (where c is a constant) vs the number of clusters, in order to look for the "elbow". We did this for 1000 training images and a range of c-values from 14million to 20million as shown in the table below. The table also includes the k-values where the "elbow" occurred. It can be seen that for c-values from 16million to 18million, the optimized k value remains the same and then changes more drastically away from this range. Therefore, we concluded that the optimized number of clusters for k-means is 25.

Constant (c)	Optimized k (at elbow)
14,000,000	30
15,000,000	26
16,000,000	25
17,000,000	25
18,000,000	25
19,000,000	21
20,000,000	19

c = 14,000,000



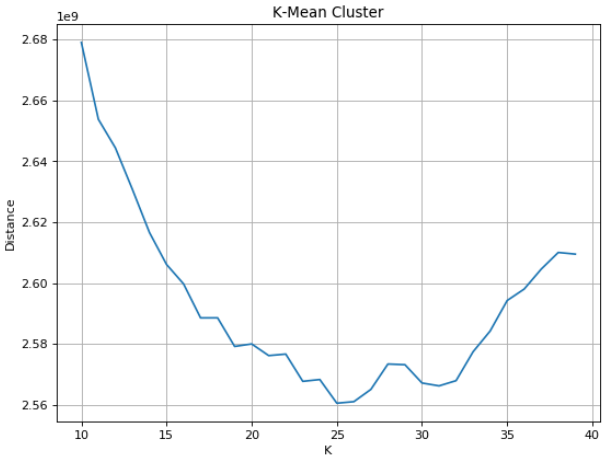
c = 15,000,000



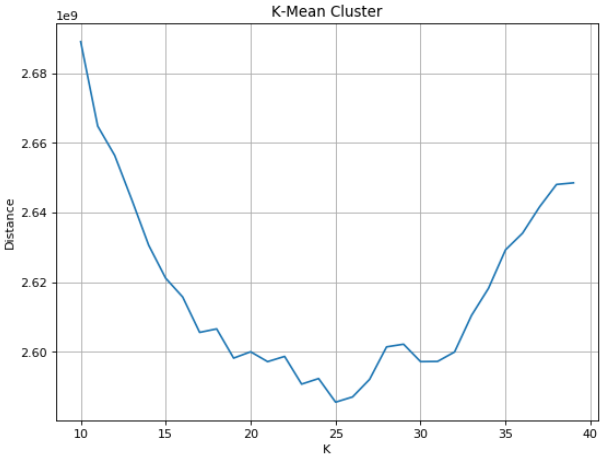
Brandon Death 101017130
Matt Walker 100980836

Nov. 25, 2018.

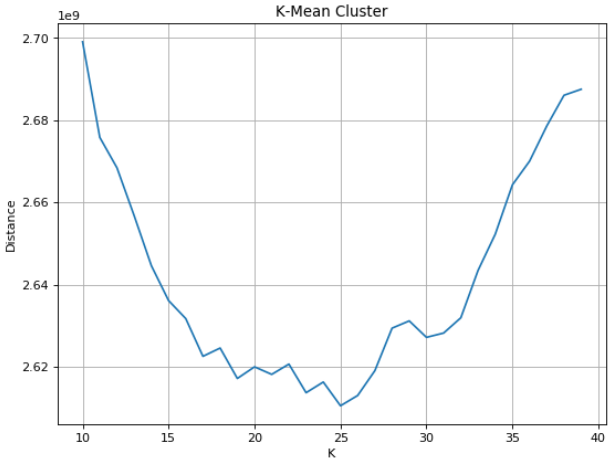
c = 16,000,000



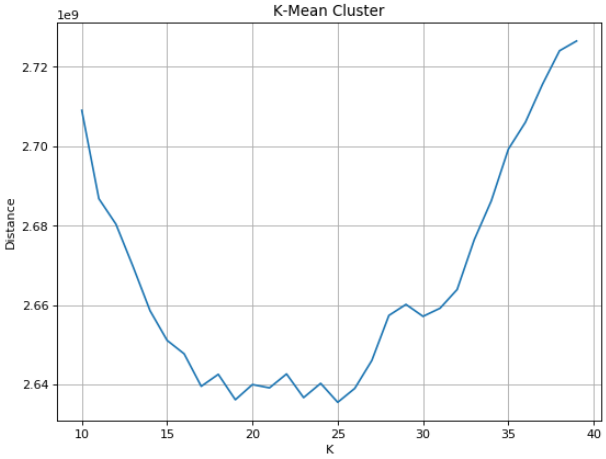
c = 17,000,000



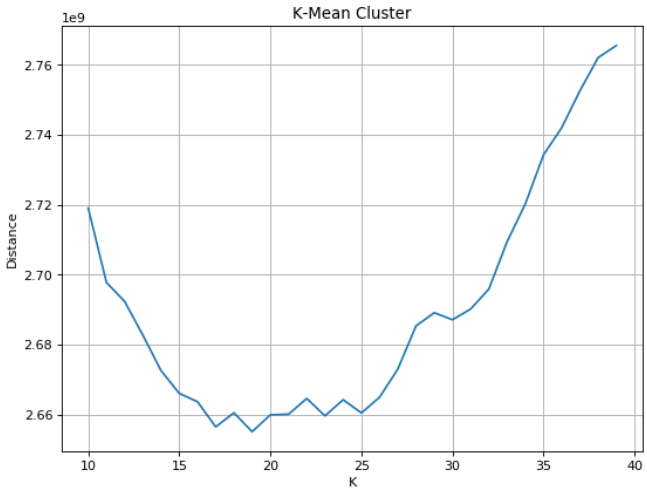
c = 18,000,000



c = 19,000,000



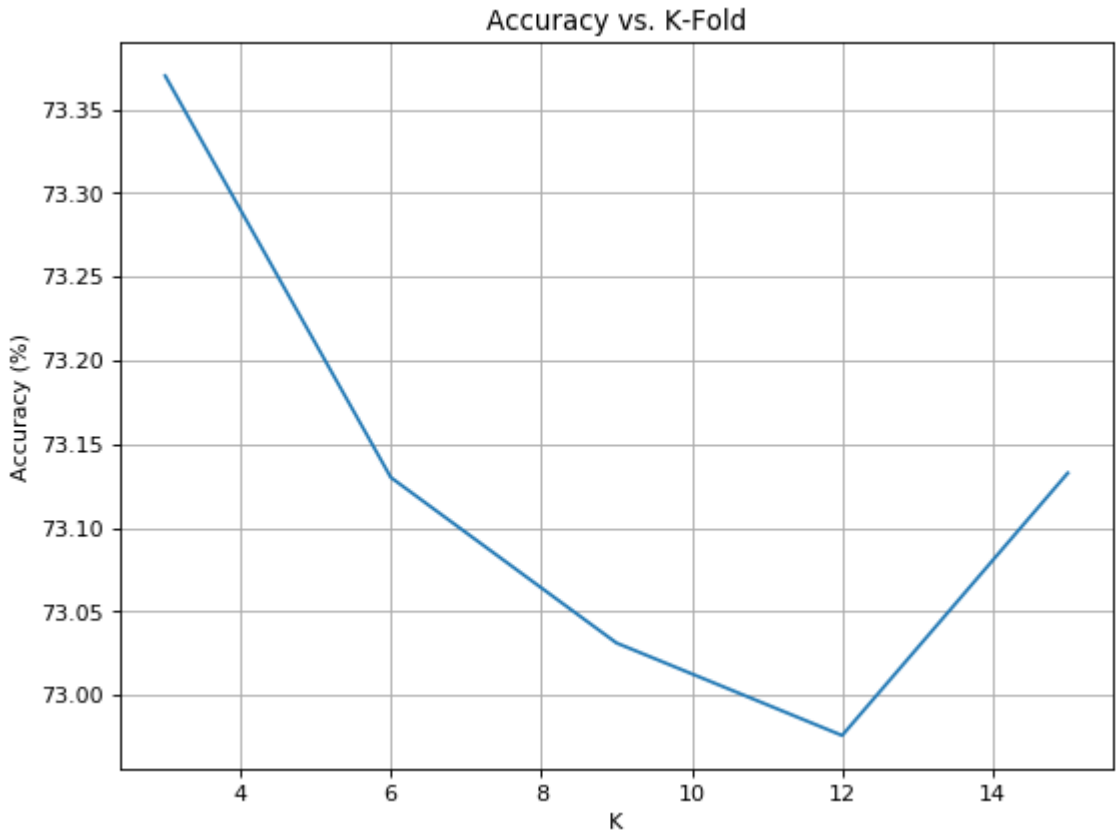
c = 20,000,000



Brandon Death 101017130
Matt Walker 100980836
Nov. 25, 2018.

2. For this question, we iterated through k-fold values of 3, 6, 9, 12 and 15, using 25 hidden layer neurons and 5000 images, to see how it affects the accuracy. The results are shown below:

K-Fold	Average Accuracy (%)
3	73.37
6	73.13
9	73.03
12	72.96
15	73.13

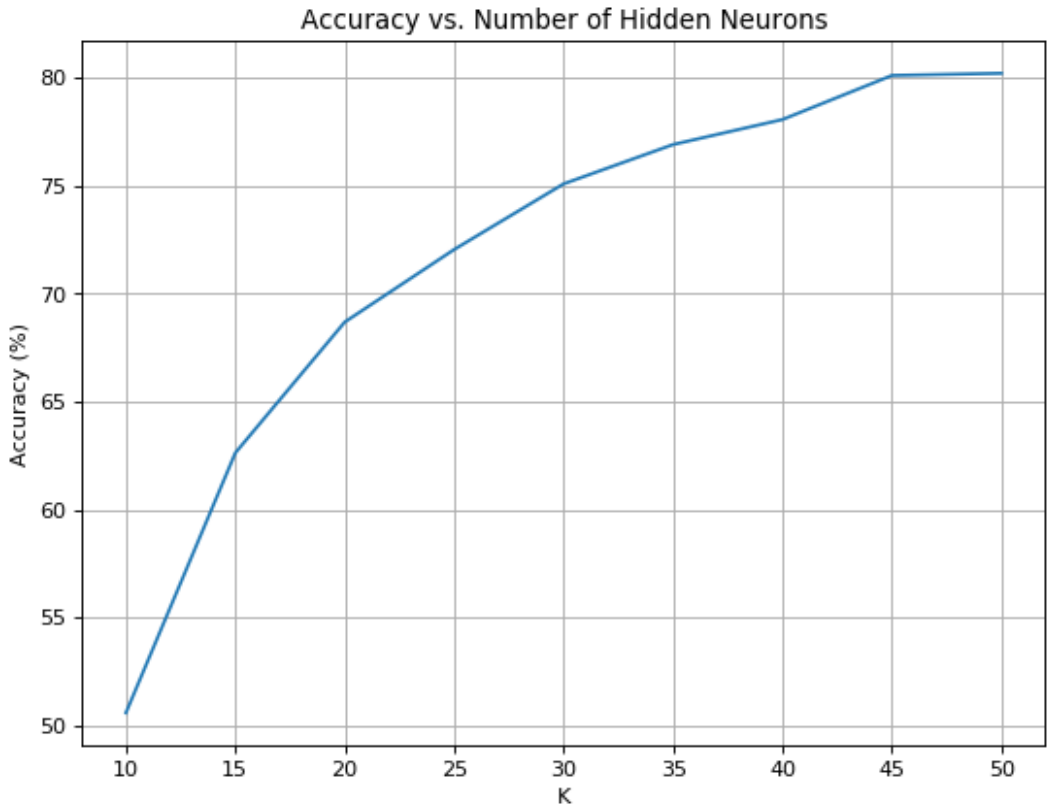


From these results, it is observed that there is very small variation in accuracy with the change in the number of k-fold.

Brandon Death 101017130
Matt Walker 100980836
Nov. 25, 2018.

3. For this question, we iterated through the 10, 15, 20, 25, 30, 35, 40, 45, and 50 number of hidden layer neurons, using 5-fold cross validation, and calculated their corresponding average accuracies. The results are shown below:

Number of Hidden Layer Neurons	Average Accuracy (%)
10	50.58
15	62.62
20	68.68
25	72.04
30	75.08
35	76.90
40	78.06
45	80.1
50	80.2

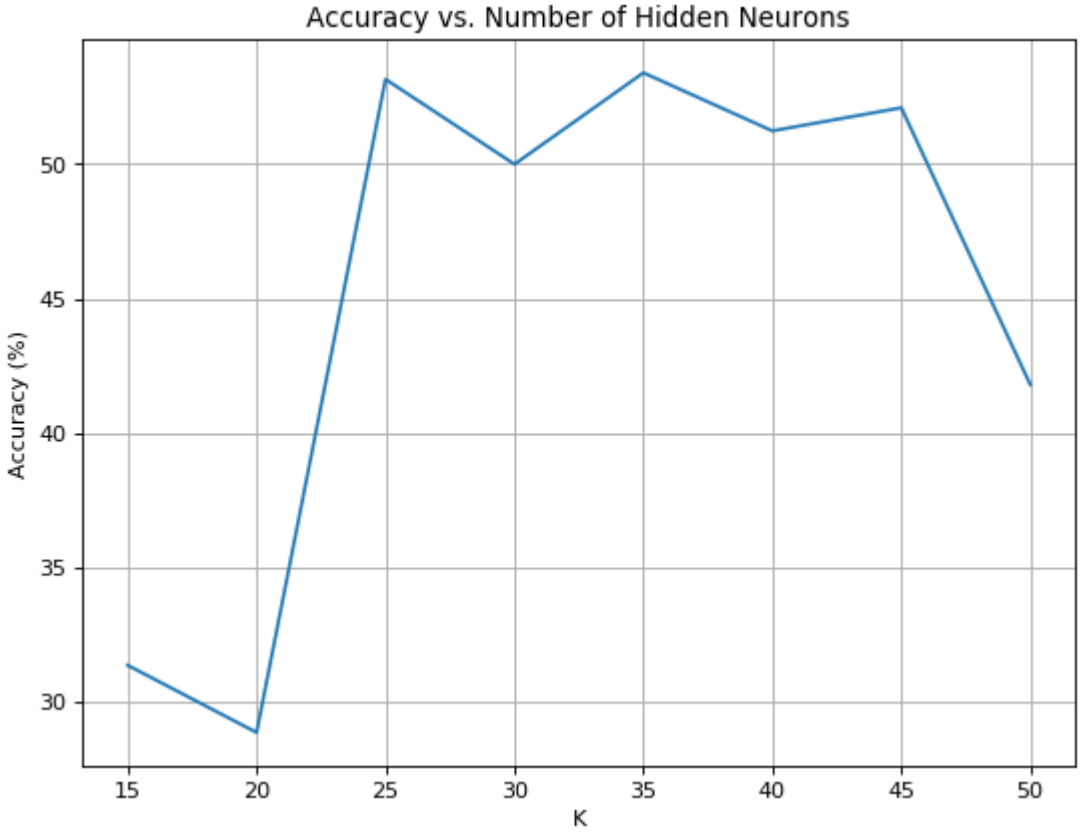


It can be seen from these results that the accuracy increases as the number of hidden layer neurons increases. The k=25 neurons, found as the optimized value from Q.2.1, gives a decent accuracy, but the accuracy is improved up to k=45 hidden neurons, where it then appears to flatten out.

Brandon Death 101017130
Matt Walker 100980836
Nov. 25, 2018.

4. For this part, we ran the same network as the previous part (3), but implemented a drop-out, where each hidden layer neuron has a 50% chance of being “shut-off”. The results are shown below:

Number of Hidden Layer Neurons	Average Accuracy (%)
15	31.36
20	28.86
25	53.16
30	50.00
35	53.40
40	51.24
45	52.10
50	41.80



The network was a lot less accurate with the drop-out implementation and plateaued at a maximum of about 55%, before decreasing as the number of hidden neurons was increased to 50.

Brandon Death 101017130
Matt Walker 100980836

Nov. 25, 2018.

Question 3

a) The SOM that we computed is a 28x28 matrix of "weights", and a learning parameter of 0.02 was used to generate it. It is shown below in part b).

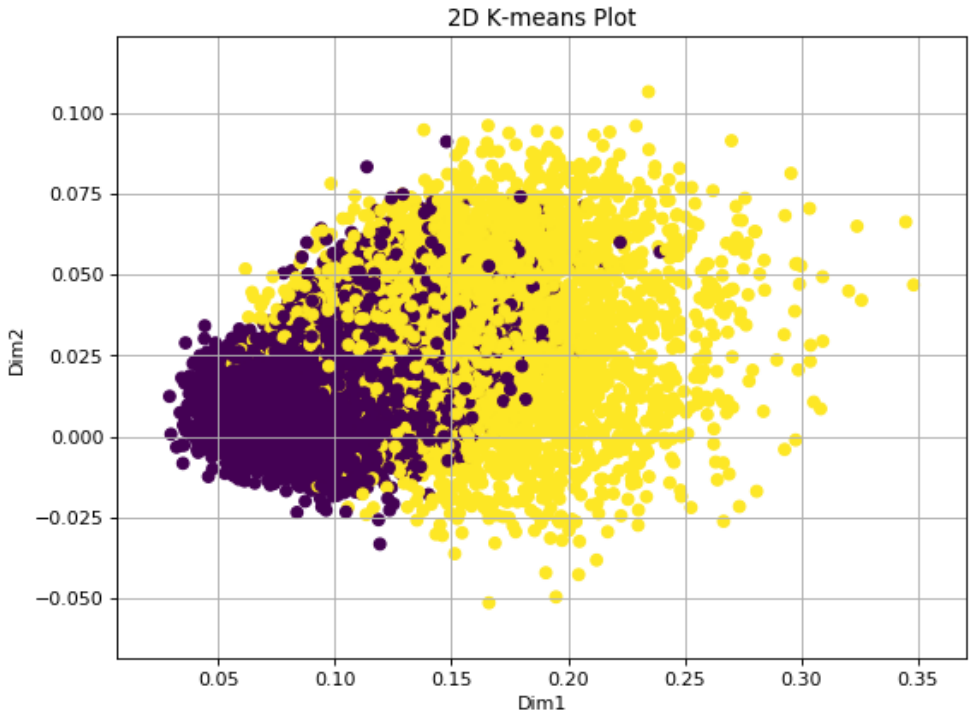
b)

```
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]]
```

The matrix above shows the cluster separation of the "weight" matrix of the SOM. The 0 and 1 values represent the separated one and five images.

Brandon Death 101017130
Matt Walker 100980836

Nov. 25, 2018.



The above plot shows the 2D representation of the training images. Each dot is a single image. It was produced by applying the SVD to the training image matrix in order to reduce the dimensionality from 784 to 2. The colouring of each point (either a 1 or a 5) was determined by clustering using kmeans.

Brandon Death 101017130
Matt Walker 100980836

Nov. 25, 2018.

Question 4

In both cases, the feed forward neural networks have 150 hidden neurons. This was determined through trial and error, which found that any less than 150 hidden neurons had reduced accuracy when used in the orthonormally fed network, and any more than 150 made little difference. The network fed with raw data does not appear to improve at all, regardless of the number of hidden neurons used.